CHAMELEON : Adaptive Fault Tolerance Using Reliable, Mobile Agents

R.K. Iyer, Z. Kalbarczyk, S. Bagchi, K. Whisnant

Center for Reliable and High-Performance Computing,
Coordinated Science Laboratory
University of Illinois at Urbana-Champaign
http://www.crhc.uiuc.edu/DEPEND

Motivation

- In contemporary networked computing systems need for applications with broad range of availability to co-exist on the same environment
- Not cost effective to develop separate platform for each type of application. Chameleon is proposed as an unified framework.
- Support wide range of criticality requirements in a dynamic and adaptive manner in a heterogeneous networked environment
- Methods and techniques in Chameleon embodied in a set of specialized, reliable and intelligent Agents supported by a Fault Tolerance Manager (FTM) and Host Daemons for handshaking with the FTM via agents

System Characteristics

- Dynamic nature
  - Handle tasks with different dependability (availability & reliability) requirements
  - Handle application having segments of varying criticality
- Adaptation to various architectures and networks
  - System specific knowledge (eg. querying system information) built into the software for a wide range of platforms
  - TCP/IP communication protocols used are widely supported over varied networks
- Rapid Error Detection & Recovery
  - Application/process monitoring : By Agents through signals
  - Agent monitoring : By Host Daemons through explicit IPC
  - Node monitoring : By FTM through heartbeats

System Characteristics (Contd.)

- Cost Effective Services
  - Components off the shelf
  - No dedicated fault tolerant architecture required of end users
  - No specialized software to be installed at user nodes
- Scalability in two major dimensions
  - Physical Scalability - System structure based on the high-speed Myrinet. Adding new computation nodes can improve system performance while communication overheads grow slowly with system size.
  - Fault Tolerance Scalability - We provide a library of basic building blocks (eg. vote(a,b,2,2,EXACT)) and agents. Both may be extended by the user or the FTM.

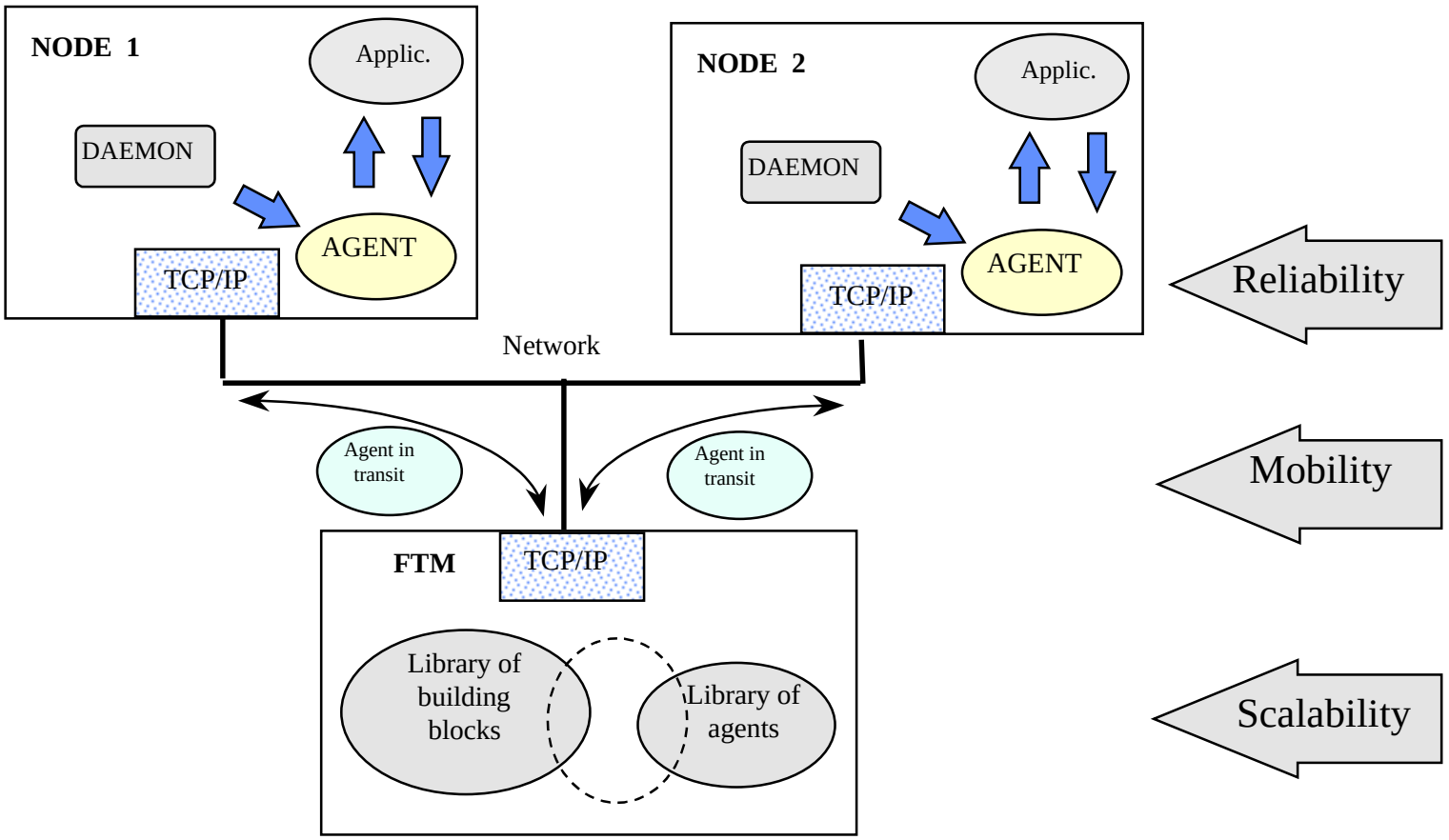
Fault Tolerance Manager (FTM)

- Centralized controlling entity for the whole environment
- Responsibility of accepting user's application and executing in the desired mode of reliability
- Can run on a dedicated fault-tolerant machine with the surrogate FTM running on a different machine
- Primary Functions
  - Mapping the network
  - User communication
  - Determining fault tolerance strategy
  - Creating and/or invoking appropriate agents for execution
  - Monitoring the system
  - Error recovery
- Three data structures at the FTM maintain information about system and current state
  - System Configuration Table, Runtime Application Table, Application Specification Table

Reliable Agents

- Failure resilient carriers of information/stimulus to/from FTM
- Designated by FTM to perform operations for successful completion of application in the designated mode
- Intelligent enough to operate autonomously
- Primary agent characteristics :
  - mobility : - migrate through the network to accomplish actions specified by FTM.
    - supported using TCP/IP communication protocols
  - reliability : - monitored by host daemons
    - rigorously tested against erroneous execution
    - fail-silent behavior ensured
  - scalability: - library of basic building blocks eg. vote, monitor
    - library of agents arranged as a hierarchy :
      - basic built-in agents
      - agents extensible from existing agents
      - agents created by user

Primary Characteristics of Agents



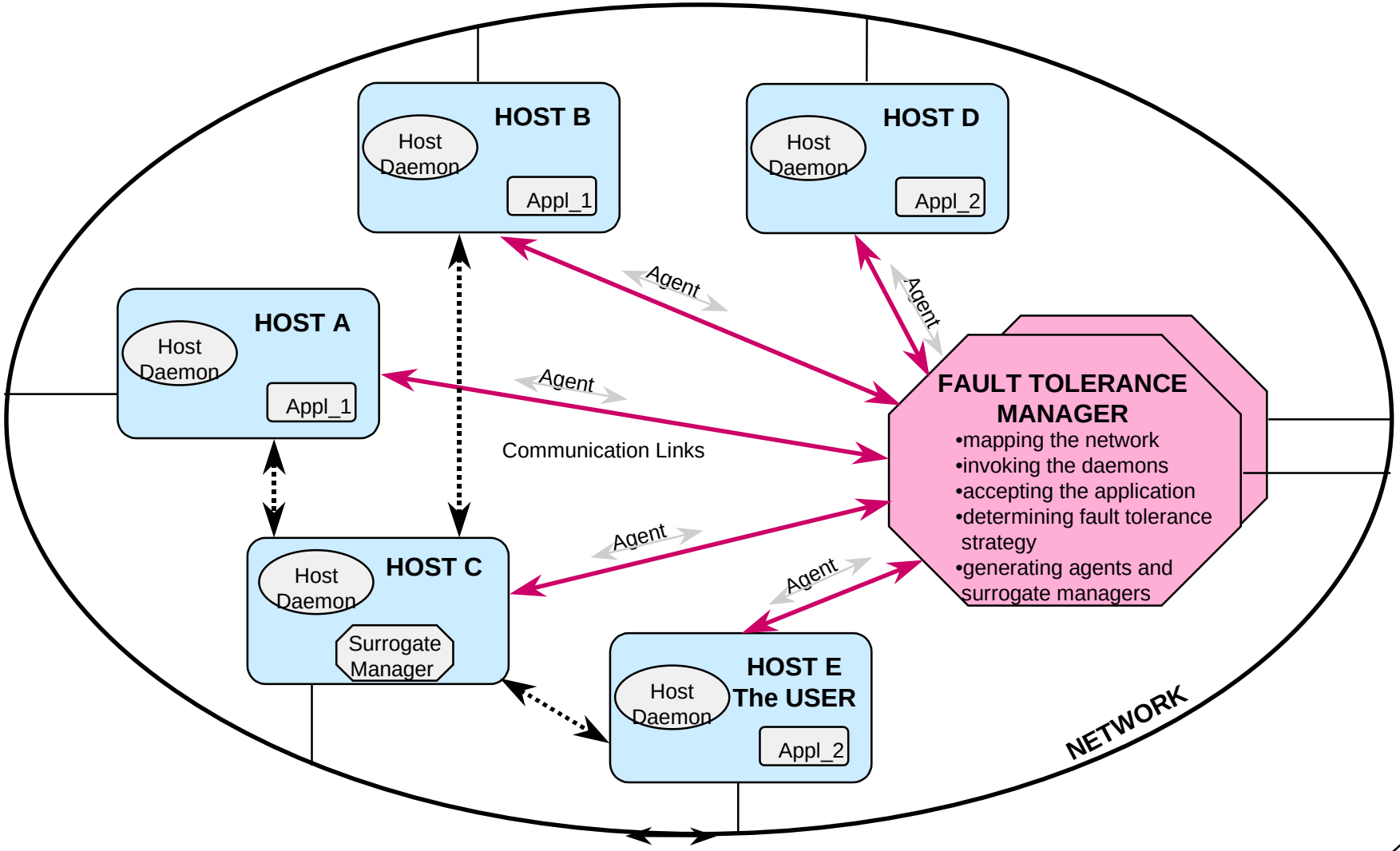
User Specified FT Strategies

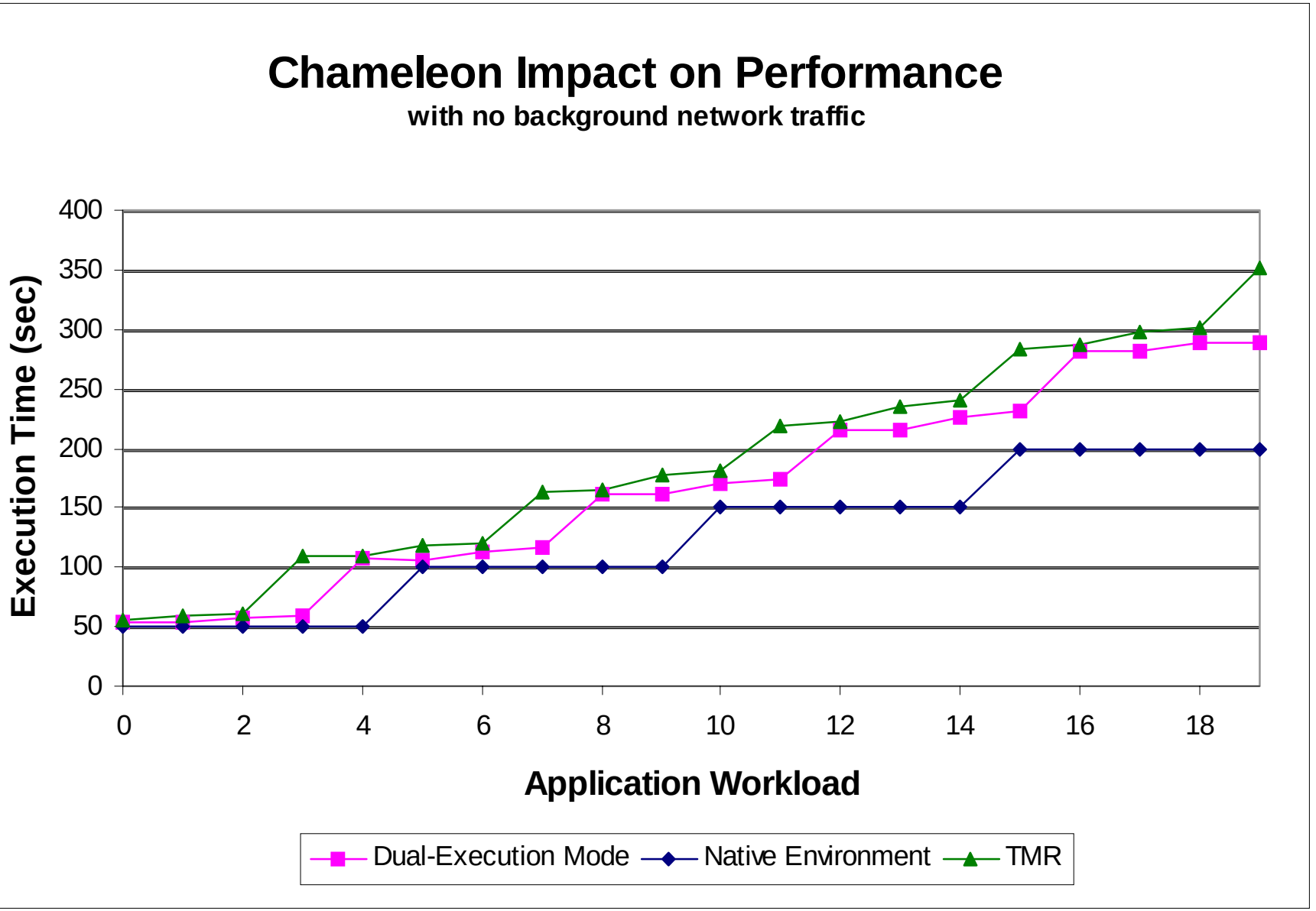
- From user's application specification, FTM can build agents to support FT strategy not built into the environment
- This scheme has limitations and can handle only a restricted set of user functionalities
- User has handle over:
  - degree of replication
  - decision mechanism
    - voting : k of n
    - acceptance : user provided acceptance function
  - agreement criterion : exact or inexact - if inexact user specified range
  - basic building blocks with user provided code that can override blocks existing in the library

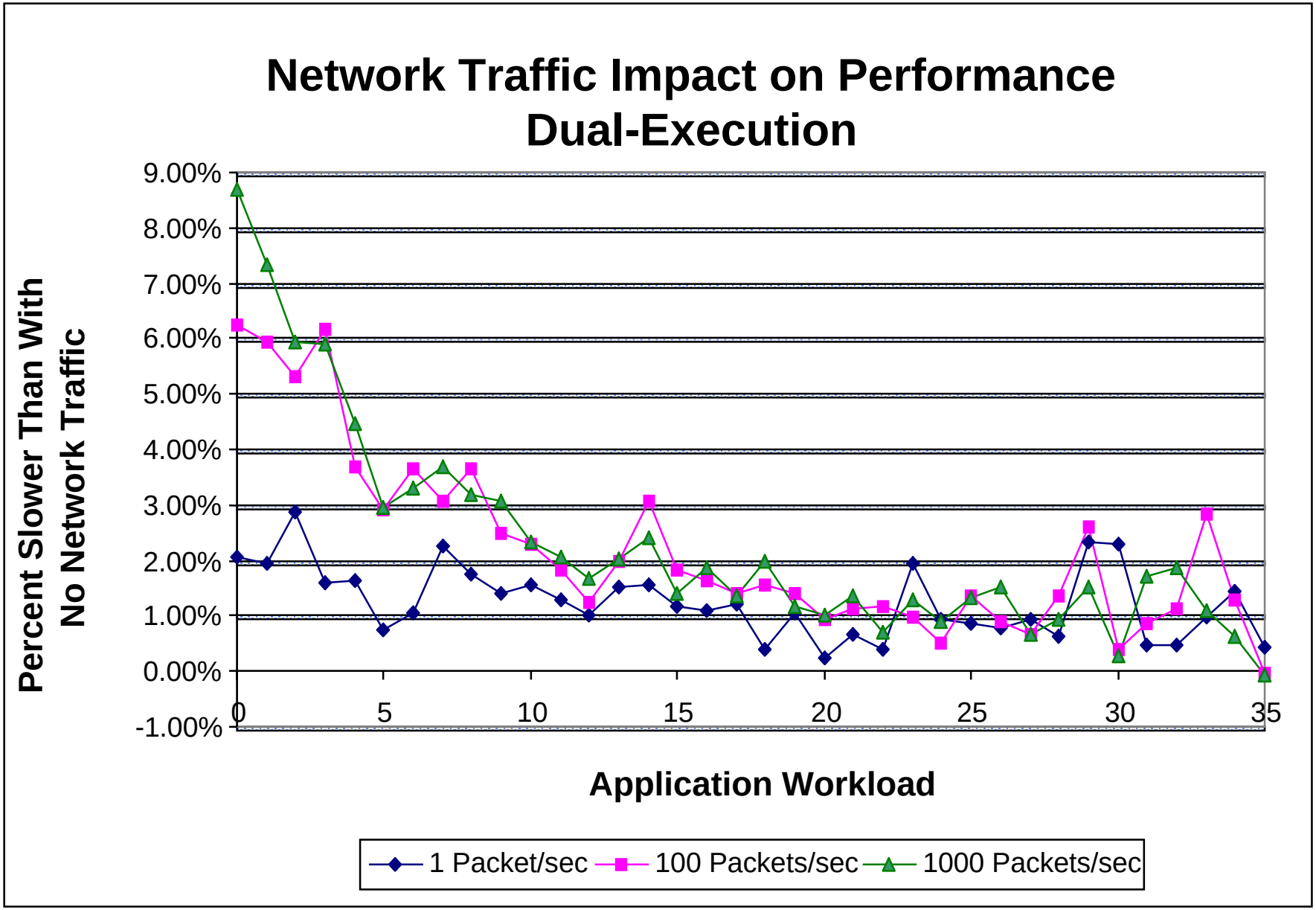
Host Daemon

- Entities residing at each host responsible for handling communication with FTM via agents
- Well-defined handshaking protocol to recognize the agents and interact with them
- Also responsible for monitoring agents and ensuring fail-silent behavior

Chameleon Infrastructure




Chameleon Impact on Performance
with no background network traffic
Execution Time (sec)
400
350
300
250
200
150
100
50
0
0
2
4
6
8
10
12
14
16
18
Application Workload
Dual-Execution Mode
Native Environment
TMR

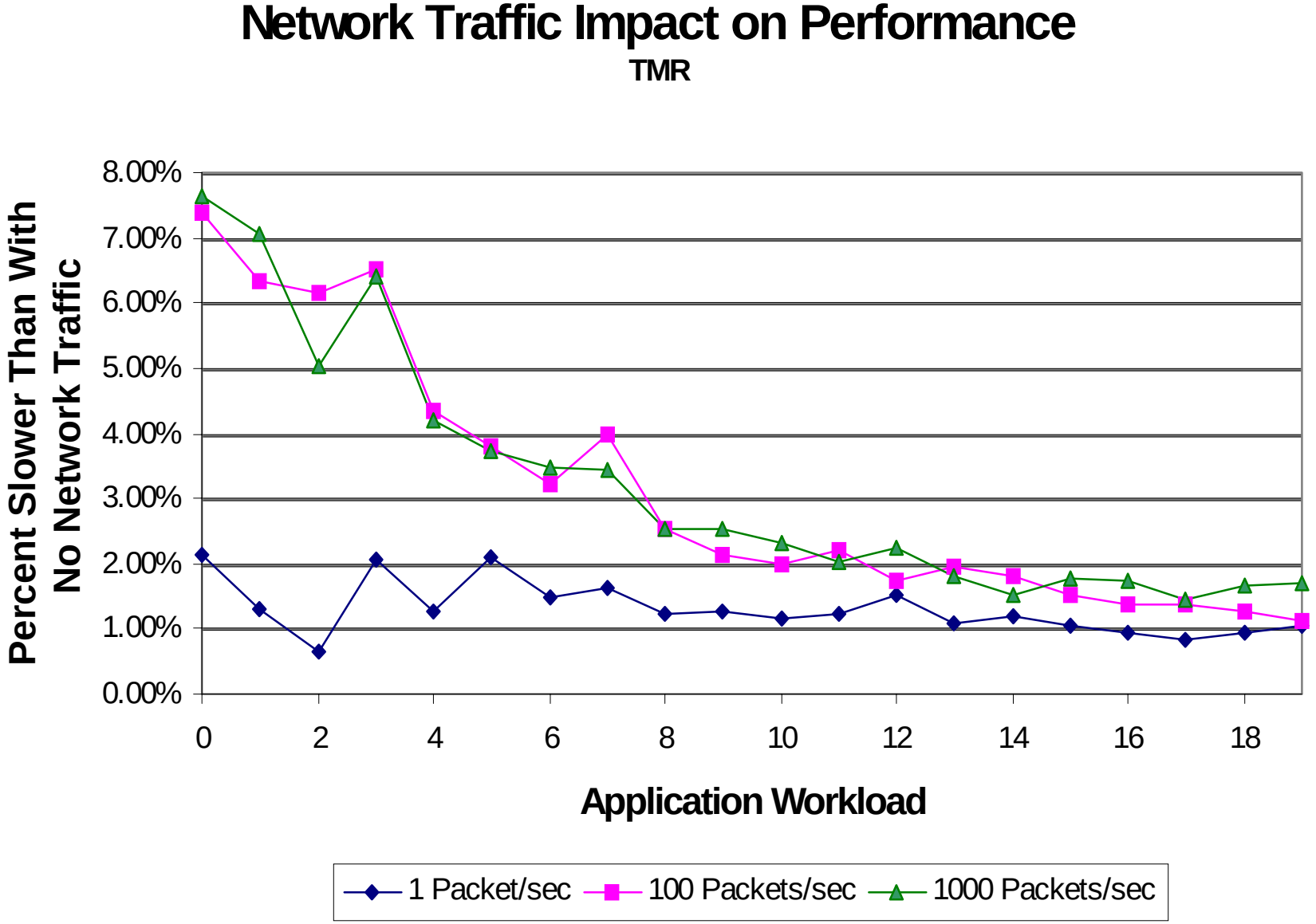

Network Traffic Impact on Performance
Dual-Execution
Percent Slower Than With No Network Traffic
9.00%
8.00%
7.00%
6.00%
5.00%
4.00%
3.00%
2.00%
1.00%
0.00%
-1.00%
0
5
10
15
20
25
30
35
Application Workload
1 Packet/sec
100 Packets/sec
1000 Packets/sec

Chameleon Performance Degradation - Dual-Execution

Application: 8K executable size, 50 sec execution time
Mode: Dual-Execution Mode
Environment: Ideal
Conditions: Constant background application workload
Constant background network traffic

App Load	Time	NO TRAFFIC Time	1 PACKET/SEC Time	1 PACKET/SEC % Change	100 PACKETS/SEC Time	100 PACKETS/SEC % Change	1000 PACKETS/SEC Time	1000 PACKETS/SEC % Change
0	50.000000	53.219031	54.319030	2.07%	56.540551	6.24%	57.844105	8.69%
1	50.000000	55.023000	56.103000	1.96%	58.284000	5.93%	59.064000	7.34%
2	50.000000	55.984000	57.594000	2.88%	58.953000	5.30%	59.303000	5.93%
3	50.000000	105.911002	107.601002	1.60%	112.434034	6.16%	112.142002	5.88%
4	100.000000	106.611002	108.362002	1.64%	110.551002	3.70%	111.362002	4.46%
5	100.000000	110.567000	111.376000	0.73%	113.777000	2.90%	113.836000	2.96%
6	100.000000	111.567000	112.726000	1.04%	115.656508	3.67%	115.247000	3.30%
7	100.000000	160.904002	164.544002	2.26%	165.814002	3.05%	166.813002	3.67%
8	150.000000	161.643002	164.444002	1.73%	167.575002	3.67%	166.794002	3.19%
9	150.000000	164.929000	167.249000	1.41%	169.039000	2.49%	169.999000	3.07%
10	150.000000	167.129000	169.709000	1.54%	170.948000	2.29%	171.039000	2.34%

Network Traffic Impact on Performance



Chameleon Performance Degradation - TMR

Application: 8K executable size, 50 sec execution time
Mode: TMR
Environment: Ideal 6-node network
Conditions: Constant background application workload
Constant background network traffic

App Load	Time	NO TRAFFIC Time	1 PACKET/SEC Time	1 PACKET/SEC % Change	100 PACKETS/SEC Time	100 PACKETS/SEC % Change	1000 PACKETS/SEC Time	1000 PACKETS/SEC % Change
0	50.000000	54.736550	55.907554	2.14%	58.772637	7.37%	58.917160	7.64%
1	50.000000	59.910000	60.680000	1.29%	63.710000	6.34%	64.130000	7.04%
2	50.000000	61.340000	61.750000	0.67%	65.120000	6.16%	64.420000	5.02%
3	50.000000	108.700001	110.950001	2.07%	115.770001	6.50%	115.680001	6.42%
4	50.000000	109.790001	111.200001	1.28%	114.560509	4.35%	114.420001	4.22%
5	100.000000	117.520000	120.000000	2.11%	121.999492	3.81%	121.920000	3.74%
6	100.000000	119.910000	121.680000	1.48%	123.760000	3.21%	124.090000	3.49%
7	100.000000	163.700001	166.340001	1.61%	170.230001	3.99%	169.330001	3.44%
8	100.000000	165.600001	167.610001	1.21%	169.820001	2.55%	169.820001	2.55%
9	100.000000	176.750000	178.960000	1.25%	180.500000	2.12%	181.230000	2.53%
10	150.000000	180.340000	182.420000	1.15%	183.910000	1.98%	184.510000	2.31%
11	150.000000	219.720001	222.390001	1.22%	224.590001	2.22%	224.140001	2.01%
12	150.000000	221.920001	225.290001	1.52%	225.790001	1.74%	226.890001	2.24%
13	150.000000	235.870000	238.430000	1.09%	240.510000	1.97%	240.170000	1.82%
14	150.000000	240.770000	243.680000	1.21%	245.140000	1.82%	244.440000	1.52%
15	200.000000	284.260001	287.290001	1.07%	288.610001	1.53%	289.270001	1.76%
16	200.000000	286.150001	288.850001	0.94%	290.100001	1.38%	291.120001	1.74%
17	200.000000	297.500000	300.000000	0.84%	301.620000	1.38%	301.790000	1.44%
18	200.000000	300.990000	303.800000	0.93%	304.770000	1.26%	305.990000	1.66%
19	200.000000	350.720000	354.410000	1.05%	354.710000	1.14%	356.740001	1.72%